

Variable projection for near-optimal filtering in low bit-rate block coders

Yaakov Tsaig, *Member, IEEE*, Michael Elad, Peyman Milanfar, *Senior Member, IEEE* and Gene H. Golub

Abstract—Recent work on block-based compression for low bit-rate coding has shown that employing a block coder within a sampling scheme where the image is downsampled prior to coding (and upsampled after the decoding stage) results in superior performance compared to standard block coding. In this paper, we explore the use of optimal decimation and interpolation filters in this coding scheme. We show that the problem of finding optimal filters for a general, unknown, “black-box” coder can be written as a separable least-squares problem in two sets of variables. We then elegantly solve this optimization problem using the *Variable Projection* method. The experimental results presented clearly exhibit a significant improvement over existing approaches.

Index Terms — block coders, deblocking, separable least-squares, variable projection.

I. INTRODUCTION

BLOCK-based coders, such as the JPEG standard for still image compression [1] and the MPEG-1 [2] and MPEG-2 [3] standards for coding of video sequences, are among the most common compression tools used for compression of visual data. Their low computational complexity along with their good performance make them a very attractive choice for many applications. A well known drawback of block coding schemes is the introduction of visually disturbing blocking artifacts in the reconstructed image. As block-based coders became increasingly more popular, the need to diminish the effects of blocking artifacts grew stronger. A great deal of effort has been invested in attempts to reduce these artifacts, while preserving the information content of the image.

Various post-processing techniques have been suggested for the reduction of blocking artifacts, based on different approaches. Among these we count techniques based on adaptive filtering [4], [5], projection on convex sets (POCS) [6], [7], [8], Markov random field modeling [9], [10] and many others (see [11] for a review). Most of these methods work well when the image is compressed at high or medium bit-rates, but the majority of them fail to produce satisfactory results when applied on images compressed at low bit-rates.

In search of a more suitable scheme for low bit-rate block coders, several authors considered the idea of downsampling before coding and subsequent upsampling at the decoder stage. Indeed, in [12], for instance, Bruckstein et al. considered downsampling an image before applying the JPEG coding algorithm, and interpolating at the decoder stage to obtain the image in full resolution. Earlier, Zeng and Venetsanopoulos [13] considered an interpolative image coding scheme based on these notions, and Jung, Mitra, and Mukherjee in [14] introduced the idea of “subband DCT”, which they later used for image resizing in the compressed domain [15]. Most recently, Dugad and Ahuja [16] proposed an elegant scheme for spatial scalability using non-scalable coders that again makes use of the down/up-sampling methodology.

This work was supported in part by the National Science Foundation Grants CCR-9984246 and CCR-9971010. Peyman Milanfar was also supported in part by US Air Force Grant AFOSR F49620-03

Y. Tsaig, M. Elad and G.H. Golub are with the SCCM Program, Computer Science Department, Stanford University, Stanford, CA 94305, USA.

P. Milanfar is with the Electrical Engineering Department, University of California, 1156 High Street, Santa Cruz, CA 95064, USA.

These approaches have several attractive properties. First and foremost, at low bit-rates there is a marked gain in performance, both in terms of PSNR and in terms of visual quality. Second, the computational complexity involved in coding/decoding is substantially reduced, since the input to the encoder is considerably smaller in size. In addition, the range of low bit-rates is expanded, allowing compression at lower bit-rates. Finally, since most of these methods do not modify the basic coding algorithm, but only apply pre/post processing, they can be used in applications where the codec is already implemented without introducing substantial modifications.

Motivated by these features, the authors of [12] specifically derived an analytical model for the JPEG encoder in order to obtain an optimality criterion on the sampling factor for a given input image. Throughout their experiments, they used fixed filters for decimation and interpolation, and did not consider the effect of different filters on the quality of the results. To the best of our knowledge, all of the papers that have taken the same approach used fixed, standard filters in the down/up-sampling process.

The main contribution of this paper is to address the issue of finding optimal filters in practice for the decimation and interpolation stages in order to achieve better performance. We shall see that the use of carefully chosen filters, based on least squares optimality, results in a significant gain in performance, both visually and quantitatively. We do not derive a specific model for the encoder, but consider it to be a black box. We view this as an advantage, rather than a shortcoming because under this framework, the results of this paper are not restricted solely to a particular block coding scheme, and can be applied to other coders as well.

The rest of this paper is organized as follows. Section II presents the algorithm for finding the optimal interpolation filter, by formulating the problem as linear least squares minimization. Section III addresses the problem of jointly optimizing both filters, which we formulate as a nonlinear, but separable least squares problem. We apply a mechanism known as *Variable Projection* to reduce this problem into a nonlinear least squares problem in one set of variables, and then demonstrate that even a sub-optimal approach results in superior performance. In section IV we provide a discussion of the results, summarize our findings and make some concluding remarks.

II. OPTIMAL INTERPOLATION BY LINEAR LEAST SQUARES

Throughout the rest of this paper, we consider the system shown in Fig. 1. An input image is convolved with a linear filter \mathbf{f} , and then downsampled by a factor k . The low-resolution image is then encoded using a block coder. At the decoder, the image is first decoded using the block decoder, then upsampled back to its original resolution and filtered by a filter \mathbf{g} to produce the reconstructed result. The authors in [12] took \mathbf{f} to be a standard anti-aliasing filter, and \mathbf{g} to be a linear interpolation kernel (hat function). The sampling factor k was chosen according to their analytical predictions. Throughout this paper, we will assume, for simplicity, $k = 2$. For a thorough discussion on choosing the optimal k , the reader is referred to [12].

Let X denote the input image, of size $m \times n$, Y denote the image after decimation (size $\frac{m}{2} \times \frac{n}{2}$), and \tilde{X} the reconstructed image after interpolation, as in Fig. 1. Our ultimate goal is to minimize

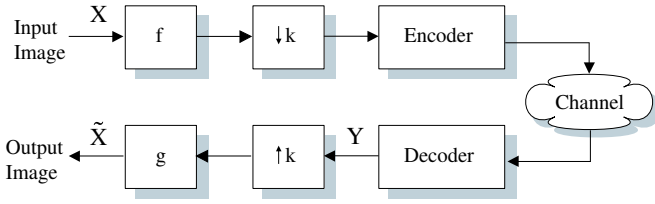


Fig. 1: Sampling/filtering scheme for block coders.

the L_2 error norm $\|\tilde{X} - X\|_2$. At first, we shall only consider the optimization of the interpolation filter \mathbf{g} , while keeping \mathbf{f} fixed. To do so, we note that the interpolation stage can be equivalently expressed as matrix multiplication. In our formulation below, we consider the upsampling and filtering steps as a unified process and consequently use a set of filters, rather than first inserting zeros and then using one filter for interpolation.

Specifically, we define $\tilde{X}^{(p,q)}$, $p, q \in \{0, 1\}$, such that

$$\tilde{X}^{(p,q)}(i, j) = \tilde{X}(2i + p, 2j + q)$$

i.e., $\tilde{X}^{(p,q)}$ is \tilde{X} shifted by (p, q) and then downsampled by 2. Clearly, the set $\{\tilde{X}^{(p,q)}, p, q \in \{0, 1\}\}$ is just a reordered version of \tilde{X} . In addition, we let $\tilde{\mathbf{x}}^{(p,q)}$ denote the row-stacked form of $\tilde{X}^{(p,q)}$, i.e. each $\tilde{\mathbf{x}}^{(p,q)}$ has length $mn/4$, with elements

$$\tilde{\mathbf{x}}^{(p,q)}(i \times \frac{n}{2} + j) = \tilde{X}^{(p,q)}(i, j)$$

Now, for our 2-D interpolation filter \mathbf{g} with dimensions $l \times l$, we similarly define $\{\mathbf{g}^{(p,q)}, 0 \leq p, q \leq 1\}$, where each $\mathbf{g}^{(p,q)}$ is a vector of length l^2 that represents the filter in the filter set which produces $\tilde{X}^{(p,q)}$ by filtering Y . Finally, we construct a matrix Φ with dimensions $\frac{mn}{4} \times l^2$ out of the image Y , of the form

$$\Phi = \begin{bmatrix} \phi_{0,0}^T \\ \phi_{0,1}^T \\ \vdots \\ \phi_{\frac{m}{2}-1, \frac{n}{2}-1}^T \end{bmatrix} \quad (1)$$

where $\phi_{i,j}$ is the row-stacked form of an $l \times l$ window, centered around the pixel location (i, j) of Y .

Using these definitions, we can now express each $\tilde{\mathbf{x}}^{(p,q)}$ (and equivalently \tilde{X}) as a product of the matrix Φ and the filter $\mathbf{g}^{(p,q)}$ in vector form,

$$\tilde{\mathbf{x}}^{(p,q)} = \Phi \mathbf{g}^{(p,q)} \quad (2)$$

Looking at Eq. (2), we can immediately see that an optimal solution (in the least-squares sense) is obtained by setting $\tilde{X} = X$ and minimizing over all $\mathbf{g}^{(p,q)}$. Specifically, we solve

$$\min_{\mathbf{g}^{(p,q)}} \|\mathbf{x}^{(p,q)} - \Phi \mathbf{g}^{(p,q)}\|_2^2 \quad (3)$$

This is a linear least squares (LS) problem, with the solution $\mathbf{g}^{(p,q)}$ given by

$$\mathbf{g}^{(p,q)} = \Phi^+ \mathbf{x}^{(p,q)} \quad (4)$$

where Φ^+ denotes the pseudo-inverse of Φ [17]. For practical purposes, one can avoid constructing the matrix Φ and its pseudo-inverse Φ^+ by applying recursive least squares (RLS), or any of its block forms.

To test the performance of this method, we used a standard JPEG coder as our block coder and applied this optimization algorithm to several standard test images. We also compared our results with those of Bruckstein *et al.* [12]. Figures 2,3, 4 show the rate distortion curves obtained for the images *Barbara*, *Goldhill* and *Boats* respectively. For

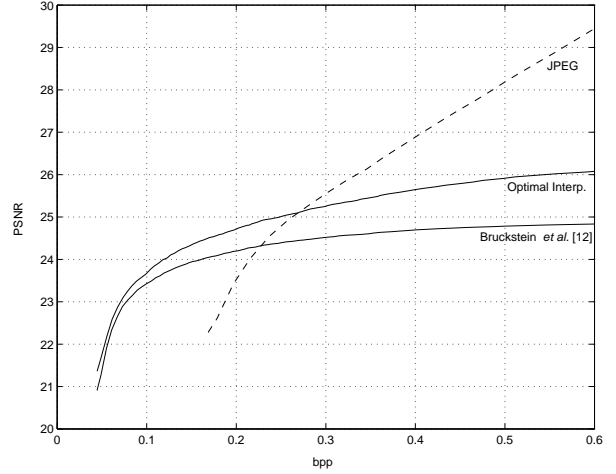


Fig. 2: Optimal Interpolation for *Barbara*.

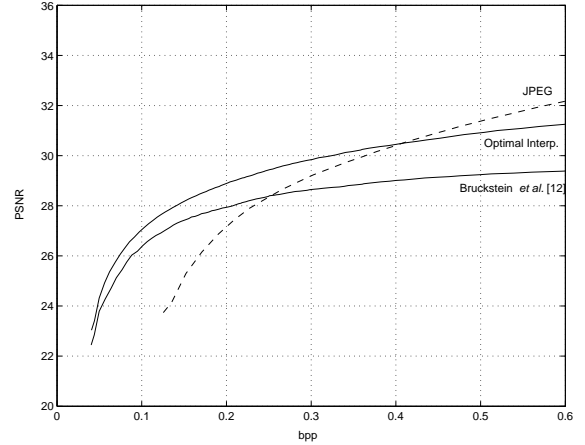
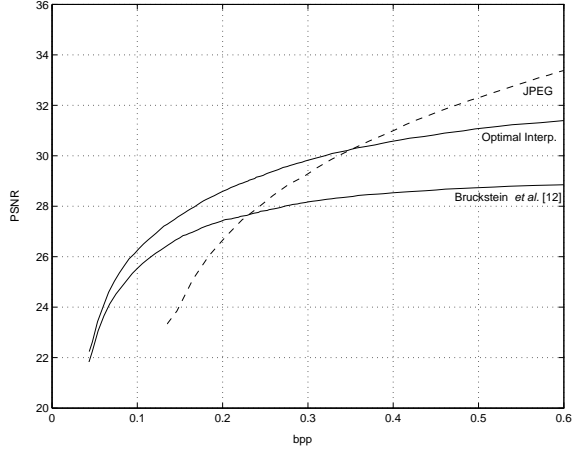


Fig. 3: Optimal Interpolation for *Goldhill*.

the decimation filter, we used an 11-tap anti-aliasing filter, designed using a Hamming window and normalized cutoff frequency $\omega_n = 0.5$. For the interpolation filter, we set $l = 5$, and used Eq. (4) to find the optimal $\mathbf{g}^{(p,q)}$. The results clearly display a significant gain in performance over the original results in [12]. We see that the optimal curve intersects the JPEG curve at a higher bit-rate than the curve obtained by Bruckstein *et al.* This essentially means that our algorithm is applicable to a wider range of bit-rates, since it performs better than direct JPEG compression up to a higher bit-rate. The visual improvement over [12] is clearly evident in figures 5, 6 and 7. We can see that the optimal filter provides a significantly sharper image, while virtually eliminating the blockiness. In terms of PSNR, in all three cases, the gain over standard JPEG is about 1.5dB.

It would be interesting to examine the filters obtained in the optimization process. Fig. 8 shows the 2-D frequency response of the optimal interpolation filters obtained for *Barbara* (recall that 4 filters are needed). As we can see, all four filters exhibit a low-pass behavior with some high-pass components. In other words, the filters attempt to smooth out the blockiness while preserving edge and texture information.

Fig. 4: Optimal Interpolation for *Boats*.

The overhead of sending the filter coefficients to the decoder is not included in the rate calculation, yet it is in the order of a few hundred bits per filter, which is negligible even for very low bit rates (we found that for the parameters used in our experiments, each filter would occupy around 200 bits when properly entropy coded). We would like to comment that one can apply the same approach using a single filter, in which case zero-padding is used prior to filtering, and only one vector \mathbf{x} is constructed. Both approaches are essentially similar; the four filters approach is a generalization of the single filter approach when sampling by a factor of 2. Our experiments show that the decrease in performance when using only one filter is rather small, and so in instances where the sampling factor is greater than 2, or when it is a rational number, a single filter can be used with a negligible loss in performance.

III. OPTIMAL DECIMATION AND INTERPOLATION BY VARIABLE PROJECTION

Inspired by the results obtained when optimizing over the interpolation filter, we now turn our attention to the decimation filter. If we consider minimizing the difference between X and \tilde{X} over both \mathbf{f} and \mathbf{g} , then the image Y at the output of the decoder is no longer fixed, hence our matrix Φ of Eq. (1) which originated from Y is now dependent on \mathbf{f} , i.e. $\Phi = \Phi(\mathbf{f})$. Nonetheless, we can still write our problem as a matrix system in the form of Eq. (2),

$$\tilde{\mathbf{x}}^{(p,q)} = \Phi(\mathbf{f})\mathbf{g}^{(p,q)} \quad (5)$$

and our minimization problem (3) now becomes

$$\min_{\mathbf{f}, \mathbf{g}^{(p,q)}} \|\mathbf{x}^{(p,q)} - \Phi(\mathbf{f})\mathbf{g}^{(p,q)}\|_2^2 \quad (6)$$

This is a non-linear LS problem with respect to the variables $\mathbf{f}, \mathbf{g}^{(p,q)}$. To find a solution for this problem, we apply Golub and Pereyra's *Variable Projection* method [18]. This method is applicable when dealing with separable nonlinear least-squares problems, i.e. problems for which the model function is a linear combination of nonlinear functions. Since first published, the Variable Projection method (sometimes coined VARPRO) has been applied in a variety of different fields in engineering and scientific computing (see [19] for an extensive survey).

In our formulation, we have two separable sets of variables, namely \mathbf{f} and $\mathbf{g}^{(p,q)}$. Moreover, the dependence on $\mathbf{g}^{(p,q)}$ is linear. Hence, we can use VARPRO to transform the problem into a nonlinear LS



(a)



(b)



(c)

Fig. 5: Compression results for *Barbara*, 0.2bpp: (a) JPEG, PSNR = 23.42dB (b) Bruckstein *et al.*, PSNR = 24.19dB (c) Optimal Interpolation, PSNR = 24.74dB.



(a)



(a)



(b)



(b)



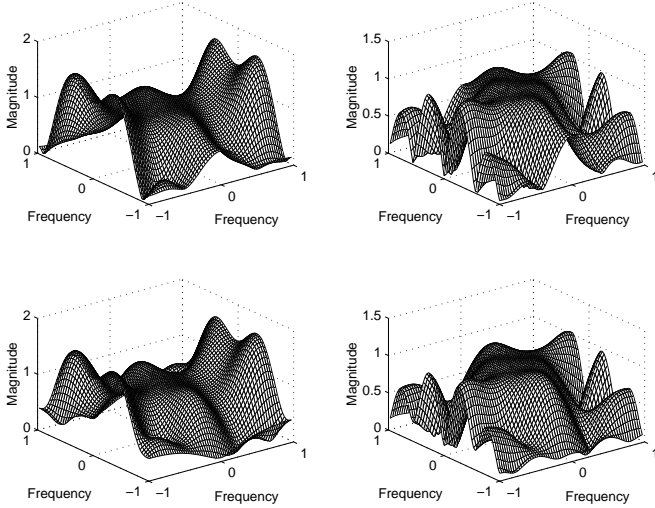
(c)



(c)

Fig. 6: Compression results for *Goldhill*, 0.2bpp: (a) JPEG, PSNR = 27.43dB (b) Bruckstein *et al.*, PSNR = 27.95dB (c) Optimal Interpolation, PSNR = 28.91dB.

Fig. 7: Compression results for *Boats*, 0.2bpp: (a) JPEG, PSNR = 26.82dB (b) Bruckstein *et al.*, PSNR = 27.42dB (c) Optimal Interpolation, PSNR = 28.58dB.

Fig. 8: Optimal interpolation filters obtained for *Barbara*

problem in one set of variables. More specifically, assume for the moment we know the optimal \mathbf{f} . If we plug this \mathbf{f} into Eq. (5), then $\Phi(\mathbf{f})$ is now fixed, hence we again face a linear LS problem, with the solution readily given by

$$\mathbf{g}^{(p,q)} = \Phi(\mathbf{f})^+ \mathbf{x}^{(p,q)} \quad (7)$$

Now, we can use this expression for $\mathbf{g}^{(p,q)}$ in our minimization problem (6), leading to

$$\min_{\mathbf{f}} \|\mathbf{x}^{(p,q)} - \Phi(\mathbf{f})\Phi(\mathbf{f})^+ \mathbf{x}^{(p,q)}\|_2^2 = \min_{\mathbf{f}} \|\mathbf{P}_{\Phi(\mathbf{f})}^\perp \mathbf{x}^{(p,q)}\|_2^2 \quad (8)$$

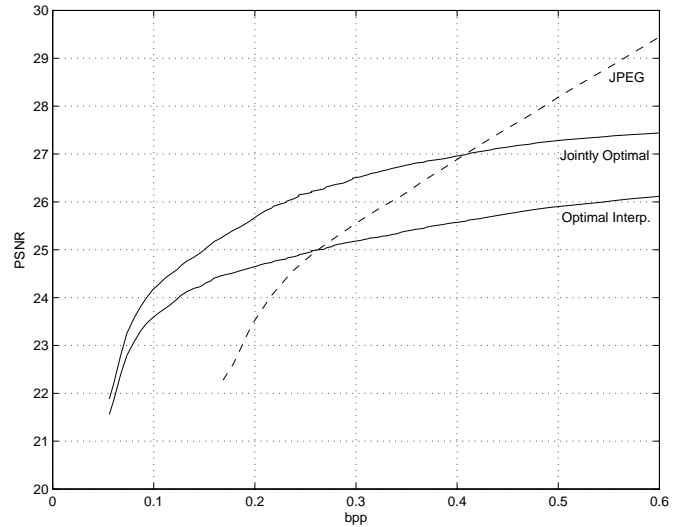
where $\mathbf{P}_{\Phi(\mathbf{f})}^\perp \equiv \mathbf{I} - \Phi(\mathbf{f})\Phi(\mathbf{f})^+$ is the projector on the orthogonal complement of the column space of $\Phi(\mathbf{f})$.

As we can see, by using VARPRO, we have essentially eliminated the minimization with respect to $\mathbf{g}^{(p,q)}$, and we are left with a non-linear LS problem with respect to the decimation filter \mathbf{f} . Evidently, this procedure not only reduces the dimension of the parameter space but also results in a better-conditioned problem (see [19] for details). Nonetheless, solving (8) is still a difficult task, due to the non-linearity of the problem at hand. For our purposes, we shall pursue a near-optimal approach, by restricting our discussion to a fraction of the parameter space for this optimization problem. We notice that the parameter space for this problem is l -dimensional, where l is the length of the filter \mathbf{f} (assuming separability of \mathbf{f}). Rather than considering the entire space, we consider a family of lowpass filters with a varying cutoff frequency. For the design of the lowpass filter $\mathbf{f}_{LP}(\omega)$, we use the window method with a Hamming window to design a separable filter with cutoff frequency ω . Consequently, a sub-optimal solution of (5) is given by the lowpass filter $\mathbf{f}_{LP}(\hat{\omega})$, where $\hat{\omega}$ is given by

$$\hat{\omega} = \arg \min_{\omega} \|\mathbf{P}_{\Phi(\mathbf{f}_{LP}(\omega))}^\perp \mathbf{x}^{(p,q)}\|_2^2 \quad (9)$$

This is a univariate minimization problem, which we solve using Brent's classic algorithm for minimization without derivatives [20]. We note that the optimization problem (9) is solved once at the encoding stage, and the resulting decimation filter need not be sent to the decoder as it is not used in the decoding process. Hence, no additional overhead is introduced by this stage.

Fig. 9 shows the rate-distortion curve obtained when applying this algorithm on the test image *Barbara*. Clearly, optimizing over both filters results in significantly better quality compared to optimization

Fig. 9: Optimal Interpolation & Decimation for *Barbara*.Fig. 10: Optimal decimation & interpolation for *Barbara*, 0.2bpp, PSNR = 25.5dB.

of the interpolation filter alone. This is also inherent in Fig. 10, which displays the visual results for a bit rate of 0.2bpp (compare Fig. 5). The image obtained from the joint optimization is sharper and exhibits more details, and is free of any blocking artifacts. Interestingly enough, the optimal decimation filter found in this case is a lowpass filter with cutoff frequency $\omega = 0.97$, which is essentially an identity filter. Intuitively, this means that we do not want any filtering prior to downsampling, in order to preserve the texture that dominates *Barbara*.

IV. DISCUSSION AND CONCLUSIONS

This paper presented an algorithm for finding optimal filters in a sampling/compression scheme. By a least squares argument, we have shown that the use of optimal interpolation filters achieves a significant gain in performance, compared to using common filters. We demonstrated that a near optimal approach for finding the decimation filter also results in a marked improvement in quality. In addition, by

using optimal filters, we outperform JPEG for a wider range of bit-rates compared to [12], making this sampling/compression scheme applicable in more diverse situations.

We note that even greater improvements can be gained by tailoring the filter optimization to the block structure of the coder. By using one set of filters for block boundaries, and another for block interior, we are likely to get better performance, as there is much to be gained by treating these two regions separately. We leave this issue for future research.

REFERENCES

- [1] *Digital compression and coding of continuous-tone still images, Part 1, requirements and guidelines*, ISO/IEC JTC1 Committee Draft 10918-1 Std., February 1991.
- [2] *Information technology - coding of moving pictures and associated audio for digital storage media at up to about 1.5 mbit/s.*, ISO/IEC 11172 Std., 1993.
- [3] *Information technology - generic coding of moving pictures and associated audio information.*, ISO/IEC 13818 Std., 1994.
- [4] C. J. Kuo and R. J. Hsieh, "Adaptive postprocessor for block encoded images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, no. 4, pp. 298–304, August 1995.
- [5] J. Chou, M. Crouse, and K. Ramchandran, "A simple algorithm for removing blocking artifacts in block-transform coded images," *IEEE Signal Processing Lett.*, vol. 5, pp. 33–35, February 1998.
- [6] D. Yula and H. Webb, "Image restoration by the method of convex projections: Part i - theory," *IEEE Trans. Med. Imag.*, vol. 1, pp. 81–94, October 1982.
- [7] A. Zakhor, "Iterative procedures for reduction of blocking effects in transform image coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, no. 1, pp. 91–95, March 1992.
- [8] Y. Yang, N. P. Galatsanos, and A. K. Katsaggelos, "Regularized reconstruction to reduce blocking artifacts of block discrete cosine transform compressed images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 6, pp. 421–432, December 1993.
- [9] T. P. O'Rourke and R. L. Stevenson, "Improved image decompression for reduced transform coding artifacts," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, no. 6, pp. 490–499, December 1995.
- [10] T. Meier, K. N. Ngan, and G. Crebbin, "Reduction of blocking artifacts in image and video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 4, p. 490, April 1999.
- [11] M.-Y. Shen and C. C. J. Kuo, "Review of postprocessing techniques for compression artifacts removal," *Journal of Visual Communication and Image Representation*, vol. 9, no. 1, pp. 2–14, March 1998.
- [12] A. M. Bruckstein, M. Elad, and R. Kimmel, "Down-sampling for better transform compression," *IEEE Trans. Image Processing*, vol. 12, no. 9, pp. 1132–1144, September 2003.
- [13] B. Zeng and A. Venetsanopoulos, "A jpeg-based interpolative image coding scheme," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Minneapolis, Minnesota*, vol. V, pp. 393–396, April 1993.
- [14] S. H. Jung, S. K. Mitra, and D. Mukherjee, "Subband dct: Definition, analysis, and applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 273–286, June 1996.
- [15] J. Mukherjee and S. Mitra, "Image resizing in the compressed domain using subband dct," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 7, pp. 620–627, July 1996.
- [16] R. Dugad and N. Ahuja, "A scheme for spatial scalability using non-scalable encoders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 10, pp. 993–999, October 1996.
- [17] G. H. Golub and C. F. Van-Loan, *Matrix Computations*, 3rd ed. Johns Hopkins University Press, 1996.
- [18] G. H. Golub and V. Pereyra, "The differentiation of pseudoinverses and nonlinear least squares problems whose variables separate," *SIAM J. Numer. Anal.*, vol. 10, pp. 413–432, 1973.
- [19] —, "Separable nonlinear least squares: the variable projection method and its applications," *Inverse Problems*, vol. 19, no. 2, pp. R1–R26, 2003.
- [20] R. P. Brent, *Algorithms for Minimization without Derivatives*. Englewood Cliffs, NJ: Prentice-Hall, 1973.